METHOD AND SYSTEM FOR PROVIDING BUSINESS INTELLIGENCE WEB CONTENT WITH REDUCED CLIENT-SIDE PROCESSING

Field of the Invention

This invention relates generally to systems and methods for a layered web-business intelligence server architecture for use in a reporting system, decision support system, business intelligence system and on-line analytical processing (OLAP) system to enable the efficient exchange of business intelligence information.

Background of the Invention

Decision support systems (DSS) have been developed to efficiently retrieve selected information from data warehouses, thereby providing business intelligence information to the organization. One type of decision support system is known as an on-line analytical processing system ("OLAP"). In general, OLAP systems analyze the data from a number of different perspectives and support complex analyses against large input data sets.

In conventional web-based OLAP access systems, the exchange of business intelligence information between World Wide Web client computer systems and business intelligence server computer systems requires substantial processing capabilities and resources on the individual client computer systems. Often, the interface between the client and the server systems require the client systems to download, install and run a plurality of web browser plugin utilities in order to view or effectively interact with the exchanged information. By requiring the client-side systems to perform a substantial

portion of the information processing, server-side applications were able to effectively manipulate the large quantities of data typically in business intelligence or OLAP environments. However, the ever changing landscape of browser and plugin software (e.g., java) makes it difficult for client-side systems to remain current with every element of technology necessary to interact with the server-side applications providing the business intelligent information. Further, for individuals who routinely use more than one client-side machine to access information, ensuring that each machine includes all required elements is difficult and sometimes impossible

Accordingly, existing business intelligence systems fail to provide a method and system for exchanging business intelligence information over a computer network wherein client-side processing and software requirements are reduced to a level compatible with virtually all client-side systems.

Summary of the Invention

The present invention overcomes the problems noted above, and provides additional advantages, by providing for a system for enabling the exchange of business intelligence information over a computer network including at least one client computer system for requesting and receiving the business intelligence information. According to the present invention, users access business intelligence information over the Internet and World Wide Web to one of a plurality of web servers that balance the incoming and outgoing load. In a preferred embodiment, the user system comprises a standard browser and communicates with the web servers without downloading any plugins or other

programs to execute reports. Web server/user communication preferably occurs through use of HTML/DHTML only without program downloading.

The web servers are then connected to one or more business intelligence server system that perform analysis using databases and data warehouses. According to a preferred embodiment, the web servers do not have a direct access to any of the databases or data warehouses on which business intelligence operations occur. Rather, the web servers pass report requests via XML to the business intelligence servers to perform the work with the databases. The business intelligence servers convert reports into XML and provide the XML-based version to the web server. The web server then transmits an HTML or DHTML report back to the client over the web.

By using such a layered architecture, a more efficient and secure system is provided. In particular, because the web server does not have direct access to the database, the system is less exposed to potential exposure to persons without authorization. Moreover, the layered architecture enables the provision of a "light" client system such that the client system only uses a browser. This is important for user devices with low processing power and memory (e.g., PDAs, WAP phones, etc.) and also reduced transmission time because the user device does not have to have a plugin downloaded with the report to be able to view the report. The system also provides better efficiency because the business intelligence server systems are higher powered and thus able to handle the processor-intensive report requests.

Other advantages of the present invention will be appreciated by those of ordinary skill in the art from the detailed description that follows.

Brief Description of the Drawings

Fig. 1 is a block diagram illustrating an architecture for a system according to an embodiment of the invention.

Fig. 2 is a flowchart illustrating steps performed by a process utilizing a query engine according to an embodiment of the invention.

Fig. 3 is a generalized block diagram illustrating one embodiment of a system for providing business intelligence information data over a computer network;

Detailed Description of Preferred Embodiments

The present invention may provide a system for web-based communication between a user and a business intelligence system. One embodiment of such a business intelligence system is shown in Figs. 1 and 2.

Fig. 1 is a block diagram illustrating a system 100 by which a variety of data resources may be accessed for business analytic, report generation and other intelligence purposes according to an embodiment of the invention. According to a preferred embodiment, the system 100 may comprise an Online Analytical Processing (OLAP) decision support system (DSS). In particular, Fig. 1 may comprise a portion of the MicroStrategy 7 or 7.1 platform which provides a preferred system in which the present invention may be provided.

In general, through using the system 100 of the invention, analysts, managers and other users may query or interrogate a plurality of databases or database arrays to extract demographic, sales, and/or financial data and information and other patterns from records stored in such databases or database arrays to identify strategic trends. Those strategic

trends may not be discernable without processing the queries and treating the results of the data extraction according to the techniques performed by the systems and methods of the invention. This is in part because the size and complexity of some data portfolios stored in such databases or database arrays may mask those trends.

In addition, system 100 may enable the creation of scheduled reports or services that are processed according to a schedule. Users may then subscribe the service, provide personalization criteria and have the information automatically delivered to the user, as described in U.S. Patent No. 6,154,766 to Yost *et al.*, which is commonly assigned and hereby incorporated by reference.

As illustrated in Fig. 1, a business, a government or another user may access the resources of the system 100 using a user engine 102. The user engine 102 may include a query input module 116 to accept a plurality of searches, queries or other requests, via a query box on a graphical user interface (GUI) or another similar interface. The user engine 102 may communicate with an analytical engine 104. The analytical engine 104 may include a set of extensible modules to run a plurality of statistical analyses, to apply filtering criteria, to perform a neural net technique or another technique to condition and treat data extracted from data resources hosted in the system 100, according to a query received from the user engine 102.

The analytical engine 104 may communicate with a query engine 106, which in turn interfaces to one or more data storage devices 108a, 108b ... 108n (where n is an arbitrary number). The data storage devices 108a, 108b ... 108n may include or interface to a relational database or another structured database stored on a hard disk, an optical

disk, a solid state device or another similar storage media. When implemented as databases, the data storage devices 108a, 108b ... 108n may include or interface to, for example, an Oracle™ relational database such as sold commercially by Oracle Corporation, an Informix™ database, a Database 2 (DB2) database, a Sybase™ database, or another data storage device or query format, platform or resource such as an OLAP format, a Standard Query Language (SQL) format, a storage area network (SAN), or a Microsoft Access™ database. It should be understood that while data storage devices 108a, 108b ... 108n are illustrated as a plurality of data storage devices, in some embodiments the data storage devices may be contained within a single database or another single resource.

Any of the user engine 102, the analytical engine 104 and the query engine 106 or other resources of the system 100 may include or interface to or be supported by computing resources, such as one or more associated servers. When a server is employed for support, the server may include, for instance, a workstation running a Microsoft Windows™ NT™ operating system, a Windows™ 2000 operating system, a Unix operating system, a Linux operating system, a Xenix operating system, an IBM AIX™ operating system, a Hewlett-Packard UX™ operating system, a Novell Netware™ operating system, a Sun Microsystems Solaris™ operating system, an OS/2™ operating system, a BeOS™ operating system, a MacIntosh operating system, an Apache platform, an OpenStep™ operating system, or another similar operating system or platform.

The data storage devices 108a, 108b ... 108n may be supported by a server or another resource and may, in some embodiments, include redundancy, such as a

redundant array of independent disks (RAID), for data protection. The storage capacity of any one or more of the data storage devices 108a, 108b ... 108n may be of various sizes, from relatively small data sets to very large database (VLDB)-scale data sets, such as warehouses holding terabytes of data or more. The fields and types of data stored within the data storage devices 108a, 108b ... 108n may also be diverse, and may include, for instance, financial, personal, news, marketing, technical, addressing, governmental, military, medical or other categories of data or information.

The query engine 106 may mediate one or more queries or information requests from those received from the user at the user engine 102 to parse, filter, format and otherwise process such queries to be submitted against the data contained in the data storage devices 108a, 108b ... 108n. Thus, a user at the user engine 102 may submit a query requesting information in SQL format, or have the query translated to SQL format. The submitted query is then transmitted via the analytical engine 104 to the query engine 106. The query engine 106 may determine, for instance, whether the transmitted query may be processed by one or more resources of the data storage devices 108a, 108b ... 108n in its original format. If so, the query engine 106 may directly transmit the query to one or more of the resources of the data storage devices 108a, 108b ... 108n for processing.

If the transmitted query cannot be processed in its original format, the query engine 106 may perform a translation of the query from an original syntax to a syntax compatible with one or more of the data storage devices 108a, 108b ... 108n by invoking a syntax module 118 to conform the syntax of the query to standard SQL, DB2,

Informix™, Sybase™ formats or to other data structures, syntax or logic. The query engine 106 may likewise parse the transmitted query to determine whether it includes any invalid formatting or to trap other errors included in the transmitted query, such as a request for sales data for a future year or other similar types of errors. Upon detecting an invalid or an unsupported query, the query engine 106 may pass an error message back to the user engine 102 to await further user input.

When a valid query such as a search request is received and conformed to a proper format, the query engine 106 may pass the query to one or more of the data storage devices 108a, 108n ... 108n for processing. In some embodiments, the query may be processed for one or more hits against one or more databases in the data storage devices 108a, 108b ... 108n. For example, a manager of a restaurant chain, a retail vendor or another similar user may submit a query to view gross sales made by the restaurant chain or retail vendor in the State of New York for the year 1999. The data storage devices 108a, 108b ... 108n may be searched for one or more fields corresponding to the query to generate a set of results 114.

Although illustrated in connection with each data storage device 108 in Fig. 1, the results 114 may be generated from querying any one or more of the databases of the data storage devices 108a, 108b ... 108n, depending on which of the data resources produce hits from processing the search query. In some embodiments of the system 100 of the invention, the results 114 may be maintained on one or more of the data storage devices 108a, 108b ... 108n to permit one or more refinements, iterated queries, joinders or other operations to be performed on the data included in the results 114 before passing the

information included in the results 114 back to the analytical engine 104 and other elements of the system 100.

When any such refinements or other operations are concluded, the results 114 may be transmitted to the analytical engine 104 via the query engine 106. The analytical engine 104 may then perform statistical, logical or other operations on the results 114 for presentation to the user. For instance, the user may submit a query asking which of its retail stores in the State of New York reached $1M in sales at the earliest time in the year 1999. Or, the user may submit a query asking for an average, a mean and a standard deviation of an account balance on a portfolio of credit or other accounts.

The analytical engine 104 may process such queries to generate a quantitative report 110, which may include a table or other output indicating the results 114 extracted from the data storage devices 108a, 108b ... 108n. The report 110 may be presented to the user via the user engine 102, and, in some embodiments, may be temporarily or permanently stored on the user engine 102, a client machine or elsewhere, or printed or otherwise output. In some embodiments of the system 100 of the invention, the report 110 or other output may be transmitted to a transmission facility 112, for transmission to a set of personnel via an email, an instant message, a text-to-voice message, a video or via another channel or medium. The transmission facility 112 may include or interface to, for example, a personalized broadcast platform or service such as the Narrowcaster™ platform or Telecaster™ service sold by MicroStrategy Incorporated or another similar communications channel or medium. Similarly, in some embodiments of the invention, more than one user engine 102 or other client resource may permit multiple users to view

the report 110, such as, for instance, via a corporate intranet or over the Internet using a Web browser. Various authorization and access protocols may be employed for security purposes to vary the access permitted users to such report 110 in such embodiments.

Additionally, as described in the '766 Patent, an administrative level user may create a report as part of a service. Subscribers/users may then receive access to reports through all sorts of data delivery devices including telephones, pages, PDAs, WAP protocol devices, email, facsimile, and many others. In addition, subscribers may specify trigger conditions so that the subscriber receives a report only when that condition has been satisfied, as described in detail in the '766 Patent. The platform of Fig. 1 may have many other uses, as described in detail with respect to the MicroStrategy 7 and 7.1 platform, the details of which will be appreciated by one of ordinary skill in the reporting and decision support system art.

The steps performed in a method 200 for processing data according to the invention are illustrated in the flowchart of Fig. 2. In step 202, the method 200 begins. In step 204, the user may supply input, such as a query or a request for information, via the user engine 102. In step 206, the user input query may be preliminarily processed, for instance, to determine whether it includes valid fields and for other formatting and error-flagging issues. In step 208, any error conditions may be trapped and an error message presented to the user, for correction of the error conditions. In step 210, if a query is in a valid format, the query may then be transmitted to the analytical engine 104.

In step 212, the analytical engine 104 may further process the input query as appropriate to ensure the intended results 114 may be generated to apply the desired

analytics. In step 214, the query engine 106 may further filter, format and otherwise process the input query to ensure that the query is in a syntax compatible with the syntax of the data storage devices 108a, 108b ... 108n. In step 216, one or more appropriate databases or other resources within the data storage devices 108a, 108b ... 108n may be identified to be accessed for the given query.

In step 218, the query may be transmitted to the data storage devices 108a, 108b ... 108n and the query may be processed for hits or other results 114 against the content of the data storage devices 108a, 108b ... 108n. In step 220, the results 114 of the query may be refined, and intermediate or other corresponding results 114 may be stored in the data storage devices 108a, 108b ... 108n. In step 222, the final results 114 of the processing of the query against the data storage devices 108a, 108b ... 108n may be transmitted to the analytical engine 104 via the query engine 106. In step 224, a plurality of analytical measures, filters, thresholds, statistical or other treatments may be run on the results 114. In step 226, a report 110 may be generated. The report 110, or other output of the analytic or other processing steps, may be presented to the user via the user engine 102. In step 228, the method 200 ends.

Now referring to Fig. 3, there is shown a generalized block diagram illustrating one embodiment of a system 300 for providing reporting system reports over a computer network in accordance with the present invention. In particular, one or more client systems 302a, 302b ... 302n (where n is an arbitrary number) that preferably execute a client browser application that supports the HTTP protocol, are connected to a computer network 304, such as the Internet. Further, one or more web server computer systems

306a, 306b ... 306n (where n is an arbitrary number), executing one or more web server applications are also coupled to the computer network 304. In one embodiment, a plurality of web server computer systems 306a, 306b ... 306n are operatively joined together to form a server cluster, thereby improving the performance of the web server applications being executed thereon.

In accordance with one embodiment of the present invention, one or more business intelligence server computer systems 308a, 308b ... 308n are operatively connected to the one or more web servers computer systems 306a, 306b ... 306n for providing the analytical and querying functions described above in relation to the system of Fig. 1. Business intelligence server computer systems may comprise an OLAP system. Also, it should be understood that business intelligence computer systems may also comprise reporting computer systems and decision support computer systems.

As with the plurality of web server computer systems 306a, 306b ... 306n, the business intelligence server computer systems 308a, 308b ... 308n may also be operatively joined together to form a server cluster. In a preferred embodiment, the one or more web server computer systems and the one or more business intelligence computer systems are configured to format, send and receive information utilizing extensible markup language (XML).

Upon receiving a query request from at least one of the client computer systems 302a, 302b ... 302n via a HTTP formatted document such as hypertext markup language (HTML) or dynamic hypertext markup language (DHTML), the web server application preferably formats the request into an XML query document utilizing a specific XML

application program interface (API). In one embodiment, this XML query document is then transmitted to at least one of the business intelligence sever applications residing on the business intelligence server computer systems 308. In an alternative embodiment, the web server application may operate to format the request into XML using a java API, rather than the XML API. As is well known in the art, java is a platform-independent, object oriented, multi-threaded and extendible programming language. The java API is essentially a simplified interface designed to enable vendors desiring to create applications for calling the business intelligence information, to format their applications without requiring a detailed understanding of the complex XML API applied. The web server, in response to receiving a java API request, completes the XML conversion for transmission to the business intelligence server computer systems.

Upon receipt of a properly formatted XML query document, the business intelligence server computer system 308 may then communicate with one or more data storage devices 310a, 310b ... 310n to execute a report to generate a report result using according to the process described above with reference to Figs. 1 and 2.

According to a preferred embodiment of the present invention, all business intelligence activity is conducted on the business intelligence server system 308 and not on the web server computer systems 306a, 306b ... 306n or the client computer systems 302a, 302b ... 302n. Indeed, according to this preferred embodiment, the webs sever systems 306 do not have direct access to any of the data in the database system used to generate the business intelligence reports. This enables efficient clustering of the web servers and results in optimized data transfer between the web server computer systems

306 and the client computer systems 302. Additionally, by limiting access to the database to the business intelligence servers, data security is substantially increased.

Once the business intelligence server system 308 has performed its logic and retrieved the requested information through use of the at least one data storage device 310a, 310b ... 310n, the entire process is reversed. In particular, the retrieved information is formatted into an XML results document, transmitted to the web server computer system 306, reformatting into a simple browser-executable format such as HMTL or DHTML and returned to the client computer system 302 via the computer network 304. By performing the processing on the clustered business intelligence server system 308, information exchange between the web server computer systems 306 and the client computer systems 304 is substantially increased.

This layered architecture is an advance over prior web-based reporting systems because it reduces the amount of processing occurring at the client end. This invention also streamlines the web-server processing for maximum performance. Also, the web servers do not have to be configured to know how to access any of the data in the databases to perform operations. This is particularly advantageous because often, a business intelligence system may have to access data from multiple different types of relational database systems. Therefore, only the business intelligence server system is provided with the information that enables it to retrieve and perform operations on data in the multiple databases.

Further, conventional systems also require a user state to be maintained at the web server. By providing the above-described layered architecture, the present invention

eliminates this requirement, thereby enabling server clustering for maximum scalability, and further requiring reduced memory expenditure per system user. By requiring the web servers to perform only the tasks associated with user navigation and html/xml conversion, the web servers are more efficiently scalable for maximum productivity.

The present invention is not to be limited in scope by the specific embodiments described herein. Indeed, various modifications of the present invention, in addition to those described herein, will be apparent to those of ordinary skill in the art from the foregoing description and accompanying drawings. Thus, such modifications are intended to fall within the scope of the following appended claims. Further, although the present invention has been described herein in the context of a particular implementation in a particular environment for a particular purpose, those of ordinary skill in the art will recognize that its usefulness is not limited thereto and that the present invention can be beneficially implemented in any number of environments for any number of purposes. Accordingly, the claims set forth below should be construed in view of the full breath and spirit of the present invention as disclosed herein.